5.3 Network layer - IP.

5.3.1 Overview. The DOD IP forms the network layer of the TACO2 protocol suite. IP provides a mechanism for transmitting blocks of data (datagrams) from sources to destinations, which are specified by 32-bit addresses. It is a "best-effort" mechanism, which provides no assurance that a datagram is delivered, but takes appropriate steps when possible to move a datagram toward its destination. IP is specified in Internet RFC 791, as amended by RFC 950 (IP Subnet Extension), RFC 919 (IP Broadcast Datagrams), and RFC 922(IP Broadcast Datagrams with Subnets). It usually also includes the Internet Control Message Protocol (ICMP), specified in RFC 792, which provides a mechanism for communicating control and error information between hosts and other hosts or gateways. Although ICMP is an integral part of IP, it uses the support of IP as if it (ICMP) were a higher level protocol. IP is also specified in MIL-STD-1777, which formally specifies a protocol consistent with RFC 791.

5.3.1.1 IP augmentations. As used in TACO2, IP may be augmented by the revised IP Security Option (RFC 1108), and by the Host Extensions for IP Multicasting (RFC 1112). These augmentations are not required in this version of TACO2, but they may be necessary for operation in certainenvironments. TACO2 supports a limited form of multicasting by allowing simplex receivers to "listen in" on simplex, half-duplex, or full-duplex transmissions. (Effectivity 5: later versions of TACO2 may support acknowledged multicast).

5.3.2 Required IP components. Because TACO2 uses IP outside its normal internetworked environment, some components of IP are unnecessary or inappropriate in some cases. This section identifies the required components for each major case.

5.3.2.1 Simplex. Simplex transmission may be used to support point-to-point or broadcast communication in TACO2. The Internet Header format shall be as specified in 5.3.3. The following fields shall be correctly filled in and interpreted for simplex operation:

a. Version

b. Internet Header Length

c. Total Length

d. Fragment Offset (must be 0)

e. Protocol (30 for NETBLT)

f. Header Checksum

g. Source Address

h. Destination Address

i. IP Security Option, if required

The remaining fields shall be disregarded by a receiver in simplex operation, but shall be provided by a transmitter for the sake of consistency. Datagrams shall not be fragmented. Subnetting support is not required. ICMP shall not be used in simplex communications.

5.3.2.2 Point-to-point duplex. Point-to-point duplex communications in TACO2 may be half-duplex or full-duplex. The Internet Header format shall be as specified in 5.3.3. The following fields shall be correctly filled in and interpreted for duplex operation:

a. Version

b. Internet Header Length

c. Total Length

d. Fragment Offset (must be 0)

e. Protocol (30 for NETBLT)

f. Header Checksum

g. Source Address

h. Destination Address

i. IP Security Option, if required

The remaining fields are not meaningful for point-to-point operation, but shall be provided by a transmitter for the sake of consistency. Datagrams shall not be fragmented. Subnetting support is not required. ICMP shall be used in point-to-point communications. However, only the following ICMP messages shall be required in this environment:

j. Parameter Problem

k. Echo

l. Echo Reply

Other ICMP messages are optional in point-to-point operation of TACO2, and shall not affect the operation of a receiver that does not implement them.

5.3.3 <u>IP Message format for TACO2</u>. Figure 17 is a summary of the contents of the internet header.

FIGURE 17. <u>Internet datagram header</u>.

a. Version: 4 bits
The Version field indicates the format of the internet header. This value shall be 4.

b. IHL: 4 bits
Internet Header Length (IHL) is the length of the internet header in 32-bit words, and thus points to the beginning of the data. Note that the minimum value for a correct header is 5.

c. Type of Service: 8 bits
Type of service choices are not meaningful in point-to-point and simplex operation. In TACO2, this field shall normally have value 0 (routine precedence, normal delay, normal throughput, normal reliability).

d. Total Length: 16 bits
Total Length is the length of the datagram, measured in octets, including internet header and data. All hosts shall be prepared to accept datagrams of up to 576 octets.

e. Identification: 16 bits
An identifying value assigned by the sender; may be ignored in TACO2.

f. Flags: 3 bits, Fragment Offset: 13 bits
TACO2 packets shall not be fragmented. A packet size shall be used that makes fragmentation unnecessary.

g. Time to Live: 8 bits
May be ignored in TACO2. This field indicates the maximum time the datagram is allowed to remain in an internet system. If this field contains the value zero, the datagram shall not be forwarded to another node.

h. Protocol: 8 bits
This field indicates the next level protocol used in the data portion of the internet datagram. The next higher level protocol used in TACO2 shall be NETBLT, which has been assigned number 30 (decimal). ICMP is protocol number 1.

i. Header Checksum: 16 bits
A checksum on the header only. The checksum field's value shall be the 16-bit one's complement of the one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero. A received datagram with an incorrect header checksum shall be discarded.

j. Source Address: 32 bits
The 32-bit IP-style address of the datagram's source. For point-to-point use, this address may be assigned arbitrarily, but shall be consistent with normal IP usage; in particular, the host portion of the address shall be neither all 1's nor all 0's (binary).

k. Destination Address: 32 bits
The 32-bit IP-style address of the datagram's destination. For point-to-point use, this address may be assigned arbitrarily, but shall be consistent with normal IP usage; in particular, the host portion of the address shall not be all 0's, and a host portion of all 1's shall be interpreted as a broadcast address. Multicast addressing per RFC 1112 may be incorporated, but is not required in this version of TACO2.

l. Options: variable
The options are optional in each datagram. If the options do not end on a 32-bit boundary, the internet header shall be filled out with octets of zeros. The first of these shall be interpreted as the end-of-options option, and the remainder as internet header padding. For the purpose of TACO2 end-points, options other than security should not be generated, but shall be tolerated if received.

5.3.4 ICMP.

5.3.4.1 Overview. Occasionally a destination host will communicate with a source host, for example to report an error in datagram processing. For such purposes the Internet Control Message Protocol (ICMP), shall be used. ICMP uses the basic support of IP as if it were a higher level protocol, however, ICMP is actually an integral part of IP. ICMP messages typically report errors in the processing of datagrams. To avoid the infinite loop of messages about messages, no ICMP messages shall be sent about ICMP messages.

5.3.4.2 ICMP in TACO2. The message formats described here are for the ICMP messages required for point-to-point duplex communications with TACO2 (see 5.3.2.2).

5.3.4.3 ICMP message formats. ICMP messages are sent using the basic IP header. The first octet of the data portion of the datagram is an ICMP type field; the value of this field determines the format of the remaining data. Any field labeled "unused" is reserved for later extensions and shall be zero when sent, but receivers shall not use these fields (except to include them in the checksum). The values of the following internet header fields shall be as described in 5.3.3:

a. Version

b. IHL

c. Type of Service

d. Total Length

e. Identification

f. Flags

g. Fragment Offset

h. Time to Live

i. Header Checksum

j. Source Address

k. Destination Address

The Protocol field shall have value 1 for ICMP.

5.3.4.3.1 Parameter problem message.

5.3.4.3.1.1 IP fields.

a. Destination Address: The source network and address from the original datagram's data.

5.3.4.3.1.2 ICMP fields.

a. Type: 12

b. Code: 0, indicating that the pointer field indicates the error.

c. Checksum: The checksum shall be the 16-bit one's complement of the one's complement sum of the ICMP message starting with the ICMP Type. For computing the checksum , the checksum field shall be zero.

d. Pointer: identifies the octet where an error was detected.

e. Internet Header + 64 bits of Data Datagram: The internet header plus the first 64 bits of the original datagram's data. This data may be used by the host to match the message to the appropriate process.

If the host processing a datagram finds a problem with the header parameters, so that it cannot complete processing the datagram, it shall discard the datagram. The host also may notify the source host via the parameter problem message. This message shall be sent only if the error caused the datagram to be discarded. The pointer shall identify the octet of the original datagram's header where the error was detected (it may be in the middle of an option). For example, 1 indicates something is wrong with the Type of Service, and (if there are options present) 20 indicates something is wrong with the type code of the first option.

5.3.4.3.2 Echo or echo reply message.

5.3.4.3.2.1 IP fields.

a. Addresses: The address of the source in an echo message shall be the destination of the echo reply message. To form an echo reply message, the source and destination addresses are simply reversed, the type code changed to 0, and the checksum recomputed.

5.3.4.3.2.2 ICMP fields.

a. Type: 8, indicating echo message, or 0 for echo reply message.

b. Code: 0

c. Checksum: The checksum shall be the 16-bit one's complement of the one's complement sum of the ICMP message starting with the ICMP Type. For computing the checksum , the checksum field shall be zero. If the total length is odd, the received data is padded with one octet of zeros for computing the checksum.

d. Identifier: an identifier to aid in matching echoes and replies, may be zero.

e. Sequence Number: a sequence number to aid in matching echoes and replies, may be zero. The data received in the echo message shall be returned in the echo reply message. The identifier and sequence number may be used by the echo sender to help match the replies with the echo requests. For example, the identifier might be used like a port in Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) to identify a session, and the sequence number might be incremented on each echo request sent. The echoer returns these same values in the echo reply.